CS 249 Final Project Proposal: Securing ROS

By: Vincent Viego

Problem Overview:

ROS is an open source robot middleware whose primary responsibility is to coordinate both publisher-subscriber and synchronous RPC-style communication amongst nodes (through topics and services respectively). Additionally, ROS provides hardware abstractions and implementations of algorithms common to a variety of robot applications. ROS is prevalent throughout industry and academia and yet, by its own admission, ROS is vulnerable to various forms of attack (e.g., man-in-the-middle, unauthorized access, etc.).

As robot capabilities continue to advance and deployment expands, one would expect to see a corresponding increase in ROS deployment and a growing desire to access robot applications remotely (e.g., one could imagine a Robots as a Service architecture). This motivates the following questions which we aim to investigate in this project:

1. What does the threat model look like for remote robot applications?
2. What security options are available and what guarantees do they provide?
3. What (performance) costs are associated with these options?

Algorithms and Systems:

To answer the questions and make progress towards solving the problem outlined above, we intend to utilize existing web security primitives in order to address ROS security. Specifically, we will augment standard ROS messaging, which utilizes TCP, with the TLS protocol in order to provide cryptographic guarantees of secrecy, integrity, and authentication. Additionally, in an effort to address Question 3, we will be benchmarking standard ROS against the security augmented ROS on a set of representative applications as well as those designed to stress the penalties imposed by TLS.

Stretch goal (based on Behzad's current work): Currently, traditional PPC robotics pipelines perform a lot of unnecessary computation by processing and propagating all data received by peripherals. However, it is likely that much of this data is so similar to previous inputs that computation over becomes redundant. Thus, it might be possible to filter ROS messages by developing some message "distance function" to reduce computational costs with minimal performance degradation. Similarly, under the assumption that real world inputs change continuously over time, we might use these "distance functions" to identify and filter sequential messages that are too different to be valid. These messages might correspond to sensor error or

maliciously injected readings, and thus removing them could increase system robustness and security. *Depending on the final scope of this project (i.e., whether or not this is a solo-project) we will investigate the use of such filters as a security feature in ROS.*

ROS Expected Behavior:

The expected behavior of the current ROS system is that it is extremely easy to infiltrate and manipulate. To demonstrate this, we plan on creating a simple proof of concept in which one node attempts to send messages to another through ROS, and an attacker with network access gains the ability to impersonate the sender, alter/stop messages, and/or manipulate messages in queues within the ROS master. We hope to demonstrate the baseline effectiveness of our security augmented ROS by demonstrating that on such a system these attacks are no longer possible.

Future Reading:

- R. White, M. Quigley, and H. Christensen, "SROS: Securing ROS over thewire, in the graph, and through the kernel," in Humanoids Workshop: Towards Humanoid Robots OS. Cancun, Mexico, 2016.
- W. Adi, "Mechatronic Security and Robot Authentication," 2009 Symposium on Bio-inspired Learning and Intelligent Systems for Security, Edinburgh, 2009, pp. 77-82. doi: 10.1109/BLISS.2009.30
- T. Denning, C. Matuszek, K. Koscher, J.R. Smith, and T. Kohno, "A spotlight on security and privacy risks with future household robots: attacks and lessons," UbiComp, 2009.
- J. Mcclean, C. Stull, C. Farrar, and D. Mascareñas, "A Preliminary Cyber-Physical Security Assessment of the Robot Operating System (ROS)," 2013, 874110. 10.1117/12.2016189.
- ROS Security Documentation

(Desired) Resources:

In speaking with Behzad earlier this week, he said that it might be possible to gain access to the MAVBench server for this project. If possible, I would like to run my security augmented ROS with resource constrained hardware-in-the-loop in order to benchmark the performance implications on a truly representative application with realistic resource constraints.